A divide and conquer approach to leads-to model checking for large-scale systems

July 28 2023

Credit: Pixabay/CC0 Public Domain

Model checking is one of the most successful computer science achievements in the last few decades. This is why Edmund M. Clarke, E.

Allen Emerson, and Joseph Sifakis were honored with the 2007 A.M. Turing Award for their role in developing model checking into a highly effective verification technology.

Model checking has been widely adopted, especially in hardware industries, as it can systematically verify a system that satisfies desired properties. However, there are still some issues to tackle in model checking, one of which is the notorious state explosion. Many techniques to mitigate the state explosion, such as partial order reduction and abstraction, have been devised.

Despite these existing techniques, they may not be sufficient to deal with the state explosion. Another goal is to increase the running performance of model checking. One promising approach to this issue is to parallelize model checking, which can make the best use of multicore architectures.

A research team from the Japan Advanced Institute of Science and Technology (JAIST), led by Professor Kazuhiro Ogata, has come up with a "divide and conquer" approach to leads-to model checking, referred to as DCA2L2MC. As indicated by the name, DCA2L2MC is dedicated to leads-to properties, which informally describe that whenever something becomes true, something else will eventually become true.

Chandy and Misra designed a temporal logic called UNITY in which the leads-to temporal connective plays an important role, and they demonstrated that many essential systems requirements can be expressed as leads-to properties. Therefore, focusing on leads-to properties is beneficial. Details about DCA2L2MC have been published in an article in *ACM Transactions on Software Engineering and Methodology*.

The core idea of DCA2L2MC is to divide an original leads-to model checking problem into multiple smaller model checking problems in a

layered way and tackle each smaller one independently. Specifically, DCA2L2MC divides the reachable state space from each initial state into L+1 layers, where L is a positive natural number, generating multiple sub-state spaces. Model checking experiments are then conducted for each sub-state space instead of the original reachable state space.

If each sub-state space is much smaller than the original reachable state space, it becomes feasible to conduct leads-to model checking, even when directly conducting it for the original reachable state space is infeasible due to the state space explosion problem. This is the key to mitigating the state space explosion problem in model checking using DCA2L2MC.

In addition, due to the nature of the divide-and-conquer approach, each smaller model checking problem can be tackled independently. Particularly, smaller model checking problems in the final layer of our division are completely independent. This is the key to improving the running performance of model checking by using parallelization for DCA2L2MC.

From the theoretical perspective, the researchers have proven a theorem that guarantees the correctness of DCA2L2MC, showing that the multiple model checking problems are equivalent to the original leads-to model checking problem. On the practical front, they have developed a support tool for DCA2L2MC in Maude, a high-performance specification/programming language based on rewriting logic. This support tool offers the flexibility to run in sequential and parallel modes as needed.

Several case studies have been conducted to demonstrate the effectiveness and efficiency of the approach in model checking leads-to properties. Furthermore, they have demonstrated that DCA2L2MC

holds significant promise as a technique for model checking leads-to properties in large-scale systems, compared to existing model checkers, such as SPIN and LTSMin.

To make the best use of DCA2L2MC, the researchers have proposed two optimization techniques: one for finding all counterexamples at once in model checking using a new model checker and another for finding a good layer configuration for DAC2L2MC using an analysis tool. The first technique plays a crucial role in generating all counterexamples efficiently in DCA2L2MC, significantly improving its running performance. The second technique is essential for finding a good layer configuration that optimizes the running performance of DCA2L2MC. By utilizing these two optimization techniques, DCA2L2MC becomes more effective and efficient in verification.

Finally, DCA2LCMC can be integrated into existing model checkers, empowering them to perform model checking on larger systems. The researchers hope that several existing model checkers will embrace DCA2LCMC as an effective and efficient technique for handling leads-to properties. Furthermore, researchers and engineers can readily adopt the technique and tool to conduct verification of systems with leads-to properties.

More information: Canh Minh Do et al, Optimization Techniques for Model Checking Leads-to Properties in a Stratified Way, *ACM Transactions on Software Engineering and Methodology* (2023). DOI: 10.1145/3604610

Provided by Japan Advanced Institute of Science and Technology